Remarks

Claims 1-6, 8-9, and 20-51 are in the application. Claims 1, 20, 27 and 33 are in independent form. Reconsideration is requested.

Claim 20 is amended to correct a typographical omission. In particular, claim 20 is amended to recite "the autorun software, upon running on the host computing device, installing or running the protected software on the host computing device." By this correction claim 20 correctly recites operation of the autorun software that is similarly recited in claim 33. Claim 33 is amended to more clearly recite operation of the autorun software.

Claim 8 is objected to over a typographical error. Claim 8 has been amended to correct the error. The Examiner notes that a paragraph in claim 1 beginning "autorun software stored…" included in a previous version of the claim filed on 8/8/2007 was omitted without proper strikethrough indications in the 4/30/2008 amendment. Claim 1 set forth above includes the paragraph beginning "autorun software stored…" with strikethrough marking to indicate deletion. This paragraph was intended to be deleted and replaced by the paragraph beginning "application launcher software stored …," which was added in the 4/30/2008 amendment. The omission of the "autorun" paragraph without strikethrough indications was inadvertent.

Claims 1-6, 8-9, and 20-40 stand rejected under 35 USC 103(a) for obviousness over US Publication No. 2003/0046447 by Kouperchaliak (hereafter Kouperchaliak) in view of Publication No. 2002/0145632 to Shmueli et al. (hereafter, Shmueli) and further in view of US Patent No. 6,829,672 to Deng (Hereafter Deng). Applicant responds as follows.

Amended claim 1 recites a protected memory component where protected software is stored so as not to be viewable or accessible by the user, "even with user password authentication," and is only accessible to be run by the application launcher software upon authentication of the application launcher software, as described in the application at paragraph [0046]:

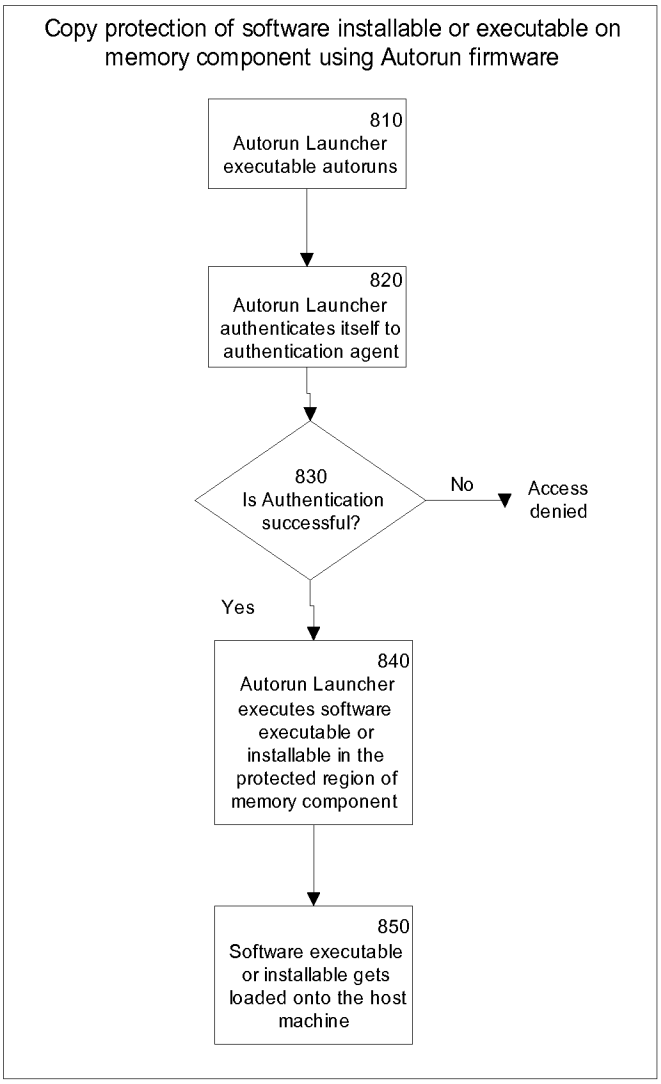
> A section of the internal memory component (e.g., memory component 620, Fig. 6) may be protected from public access by password protecting it or by physical security means such as a lock, among other means. The flash memory component can also be segmented into public and private sections. Private sections can be used to store installable or executables that cannot be viewed or accessed by the user, and public sections can be viewed or accessed by users in a conventional manner. The installable or executable software being distributed through the memory component can be stored in the protected region of the memory component. Security by way of copy protection of this installable software can be achieved by allowing only an application launcher executable, which is autorun from the memory component, to access the installable software.

As described at paragraph [0046], the present application describes a user password protection or authentication that can protect the public or internal sections of the memory component from public access or access by unauthorized users. A user can access and view such an internal memory component upon password authentication, but anyone else (i.e., any public user) could not access or view the memory component without the password authentication. The password authentication functions to authenticate the user and distinguish him or her from anyone else, thereby protecting the public sections of the internal memory component from access by other users or by the public.

However, present application also discloses of a private (or protected) memory section that "cannot be viewed or accessed by the user," and only an application launcher executable, which is autorun from the memory component, can access the installable software. These statements describing two distinct features are unambiguous. For private or protected memory, the user has no access to the private (or protected) memory section, with or without password authentication. Only the application launcher executable has access to the private (or protected) memory section, and only if the application launcher executable is authenticated. As described in paragraph [0046], the private (or protected) memory section is contrasted with password protection of an internal

or public memory section that can be viewed or accessed by the user. The present amendment of claim 1 emphasizes this distinction.

Preventing a user from accessing or viewing the private (or protected) memory section is described in the application as providing protection from copying (i.e., copy protection). Preventing access or viewing of the private (or protected) memory section denies any user the ability to see or read the contents of the private (or protected) memory section, thereby denying the user the ability to copy the contents. The copy protection functionality of the private (or protected) memory section and the accessing of it only by the application launcher executable are described with reference to Fig. 8 of the application, which is reproduced below. Only upon authentication of the launcher executable is the software in the private (or protected) memory section installed on the host computer. A user cannot access or view the private (or protected) memory section, even with password authentication. Only the launcher executable can access the software in the private (or protected) memory section.



As shown in Fig. 8 of applicant's application, there is no authentication of a user to provide access to the private (or protected) memory section because the essence of the copy protection is preventing a user from viewing or accessing software (or data) stored in the private (or protected) memory section. There can be no copy protection if a user is allowed to access or view a memory section, even with password authentication.

Kouperchaliak is directed to providing improved "plug & play" functionality of USB computer peripherals by allowing a USB peripheral to install the drivers needed to operate with a host computer. For example, Kouperchaliak describes

a computer peripheral device such as a printer that has stored on it "device-related software (DRS)" (e.g., software drivers) that permit interaction between the printer and the computer. The printer checks whether the device-related software (i.e., driver) is already installed on the host computer and, if not, uploads the device-related software from a mass storage device emulator to the computer for the proper installation and operation of the peripheral device by the computer.

Shmueli describes a portable device or "key" capable of interacting with a computing device to facilitate user interaction [Paragraph 002], the software on the portable device runs an authentication routine on host computing device via an interface to receive authentication indicia from the user [Paragraph 0011] to ensure that the user or holder of the key is authorized to use it on a host computer. The authentication routine provides a user authentication interface that requires a user to input a password, logon information, or biometric indicia from a biometric reader associated with the host 12. The user provides the authentication indicia to the interface running on the host, the authentication routine determines if the user is authenticated and provides access to data on the key to the user accordingly.

Shmueli is directed to a device that applies user authentication to allow use of the device. Shmueli provides no teaching or suggestion of a protected software that is stored in a protected memory component and is not viewable or accessible by the user, even with password authentication. Also, Shmueli provides no teaching or suggestion of a protected software that is only accessible to be run by the application launcher software upon authentication of the application launcher software, as recited in the claim. Shmeuli describes the key 10 as follows:

> "the software on the portable device may provide an authentication routine instructing the host computing device to receive authentication indicia from the user via an interface on the host computing device and determine if the authentication indicia received from the user matches authentication indicia stored on the portable device. As such, a user must

> be authenticated prior to using the portable device." Shmueli, Paragraph [0011], emphasis added.

Once he is authenticated by entry of his password, the user of a Shmueli key 10 has access to view or copy any software stored on the key 10. Accordingly, Shmueli provides no indication of a protected memory component where software is not viewable by the user. Shmueli further emphasizes the absence of any such protected memory area by relying upon user authentication or encryption for the protection of data stored on the key 10: "data may be accessed from the key 10 as necessary based on the keylet and the authentication." (Shmueli, paragraph [0039].)

The Examiner acknowledges that Kouperchaliak and Shmueli fail to teach or suggest the private (or protected) memory section recited in the claims:

> **Neither** Kouperchliak nor Shmueli explicitly disclose protected memory component. However, Deng discloses in an analogous computer system protected memory component (col. 6, lines 54-65).

Applicant notes that in the passage of Deng cited by the Examiner actually states as follows:

> The write protection pin WP has hardware write protection function, that is, it can physically protect the contents of the flash memory from being modified or erased. On the other hand, driver and firmware provide software write protection function for the external storage device. When the WP pin is at the write protection status (WP pin is connected to ground), the firmware notifies this status to the driver and the driver in turn notifies this status to the operating system. As a result, the contents in the flash memory can not be modified or erased and the data saved by the users can be protected. Especially in this case, the external storage device is impossible to be infected by virus.

It is clear from this passage that Deng is directed to write protection, which relates to protecting data from being modified or erased. Deng makes no mention or suggestion of preventing a user from accessing or viewing the contents of a memory component, which may be referred to as viewing

protection. The write protection of Deng is distinct from and independent of the viewing protection recited in the claim.

Claim 1 is explicitly directed to a memory component where protected software is stored so as not to be viewable or accessible by the user, "even with user password authentication." As stated by the Examiner, neither Kouperchaliak nor Schmueli describes a protected memory component that cannot be viewed or accessed by the user. Deng is directed to a write-protected memory that protects data from being modified or erased. Deng makes no mention of or reference to a memory that is not viewable or accessible by the user, even with user password authentication. Rather, Deng explicitly states that the "data of the flash memory can be randomly or sequentially read and written." (Deng, col. 5, lines 23-25.) Deng describes the operation of a reading command in detail, beginning at col. 5, line 45. Deng makes no mention of or reference to a memory that is not readable (i.e., viewable or accessible) by a user.

In addition to the above distinctions, neither of the cited references describes that "protected arbitrary software in the protected memory component is only accessible to be run by the application launcher software upon authentication of the application launcher software". The Examiner states that Shmueli describes authentication of an application launcher executable, as recited in the claim:

> However, Shmueli discloses in an analogous computer system *whereby the arbitrary software in the memory component cannot be viewed or accessed by the user and is only accessible to be run by the application launcher software upon authentication of the application launcher software* (paragraph [0011] "...the software on the portable device may provide an authentication routine instructing the host computing device to receive authentication indicia from the user via an interface on the host...determine if the authentication indicia received from the user matches authentication indicia stored on the portable device...user must be authenticated...")

Applicant notes that paragraph [0011] of Shmueli is explicitly directed to authentication of the user and not to authentication of the application launcher. Shmueli emphasizes that a user must be authenticated prior to using the portable device." (Shmueli, paragraph [0011].) Shmueli does not teach or suggest protected arbitrary software that is only accessible to be run by the application launcher software upon authentication of the application launcher software. As indicated above, allowing access to a memory component based on user authentication undermines the copy protection that is described in the present application. Kouperchaliak and Deng are silent as to authentication and, more particularly, are silent as to authentication of application launcher software.

Accordingly, applicant submits that none of the cited references teaches or suggests the subject matter of claim 1, particularly a protected memory component that cannot be viewed or accessed by the user, even with user password authentication and authentication of an application launcher executable. Applicant submits, therefore, that claim 1 is patentably distinct from the cited references and requests that the rejection be withdrawn.

Amended independent claims 20 and 33 include features that are recited in claim 1 and are not described in the cited references, including a protected memory component that cannot be viewed or accessed by the user, even with user password authentication and authentication of an application launcher executable. Applicants request, therefore, that claims 1, 20, and 33, and their respective dependent claims, be allowed.

Independent claim 27 recites:

> a user operable manual switch on the integrated circuit memory device that allows a user to select from among plural operating states that include a first state in which the autorun software is operable and a second state in which the autorun software is not operable so that the integrated circuit memory device functions as a conventional integrated circuit memory device.

The user operable manual switch is described in the application and shown in Fig. 7 as a mechanical structure that a user can physically manipulate by hand:

> The implementation options also include mechanisms for allowing the autorun feature to be enabled or disabled by an external mechanism (e.g., switch) that is included on the device or peripheral. The switch could be manually operable by a person. The switch could be a simple two-mode (e.g., autorun on/off) switch or could be a switch that selects from among more than two modes. (Application, paragraph [0042].)

The rejection of claim 27 is based on the same rationale as the rejection of claim 5, in which the Examiner states:

> The rejection of claim 1 is incorporated and further, Kouperchliak discloses:
>
> The integrated circuit flash drive memory device of claim 1 further comprising a user operable manual switch that allows a user to select from among plural operating states that include a first state in which the application launcher software is operable and a second state in which the application launcher software is not operable so that the integrated circuit flash drive memory device functions as a conventional integrated circuit flash drive memory device (paragraph [0037] "...Within the memory is preferably stored a series of device-related software items, each one appropriate to a different operating system or version... provided one or more configuration files allowing the peripheral device to be configured in different ways either selectable by the user or by the software").

The passage cited by the Examiner explicitly refers to "a series of device-related software items" that are selectable by the user. Applicant submit that nothing in Kouperchaliak teaches or suggests a user operable manual switch on the integrated circuit memory device. Instead, Kouperchliak discloses that a user selects from among the "series of device-related software items". The clear implication is that the user selects from among the "series of device-related software items" through a software interface, as is described in paragraph [0034]:

> If no suitable device-related software is found the user may be invited to insert a disk containing the device-related software If a

> suitable device-related software is found the user may be asked to confirm that the device-related software is suitable or to choose another device-related software, and, once a suitable device-related software is settled on, the operating system installs the device-related software and sets up the peripheral device for use

As a result, Kouperchaliak is directed to a user controlling operation of a USB computer peripheral via user interface instructions displayed on the associated computer and not a user operable manual switch on the integrated circuit memory device. Applicant submits, therefore, that claim 27 is patentably distinct from the cited references and request that the rejection of claim 27 and its dependent claims be withdrawn.

Applicant believes the application is in condition for allowance and respectfully requests the same.

IPSOLON LLP
111 SW COLUMBIA #710
PORTLAND, OREGON 97201
TEL. (503) 249-7066
FAX (503) 249-7068

Respectfully Submitted,

/Mark M. Meininger/

Mark M. Meininger
Registration No. 32,428